

Some Principles in Designing Cooperative Architectures of Autonomous Agents

Lorenzo Sommaruga, Nadia Catenazzi

Intelligent Robotics Group, Engineering Dept.,
 Universidad Carlos III de Madrid, Spain
 Butarque 15, 28911 Leganés MADRID
 tel. +34-1-624 9416, fax: +34-1-624 9430, e-mail: los@ing.uc3m.es

This work introduces some principles for designing agents architectures, which provide design guidelines concerning agent and group structuring, communication, decision making, and computational aspects for agent cooperation. They generate requirements and constraints for cooperative agents development.

The principles have been derived from a long experience in designing and implementing architectural frameworks for agent cooperation, the CooperA project (Sommaruga et al 1995). In particular, basic characteristics of autonomous agents have been formalised in a computational model.

This model consists of two elements: the agent itself and its environment (Figure 1). The model of an agent is expressed as a dynamic system in a situation (T) with inputs (I) and outputs (O), a control function (Sel) for the selection of the best action to do according to a set of cooperative behaviours (B), and an input function (In), which changes the current situation according to the input. In the environment, we distinguish two general sets, common for a whole group of agents, and independent from the agent: R and C. R is the real world, which can receive outputs from the agents, e.g. the execution of an agent action, and can provide inputs to the agents, e.g. data/observations of the real world. C represents the communication channel between the agents. The constructive aggregation of agents within an environment generates a cooperative architecture. This computational model has been abstracted in order to define a formal reference model for our principles. Some of the principles are dictated by very general constraints on computational (intelligent) systems. Others are influenced by more specific DAI and cooperative requirements. Due to the limited space, it is not possible to present here the complete formalisation, and the principles will be only mentioned. They include: the Communality Requirement (P.1), i.e. the need for overlapping skills and needs among agents; Action Selection (P.2), which maps a situation and a behaviour of an agent to a new situation with a possible output action; Communication Requirement (P.3), which states the usefulness of a communication protocol, a common dictionary, and a message passing mechanism; Group Dynamics (P.4), which allows the composition of the set of skills and needs of the whole group of agents to be varied; Self Initialisation (P.5), which permits to integrate a new agent into a group and hence makes the group of agents open; Principle of Patience (P.6), expressing the agent ability to wait for some change; Goal

Induction (P.7) of the possible agent actions, based on an act-based semantics; Cooperative Heuristics (P.8), which express programmable micro-behaviours of an agent, and are explicitly defined in the form of rules; Compositionality of Local micro-behaviour (P.9), which determines the global agent behaviour; and Compositionality of Global Behaviour (P.10) of a group, which is determined by the composition of the behaviours of each agent in the group.

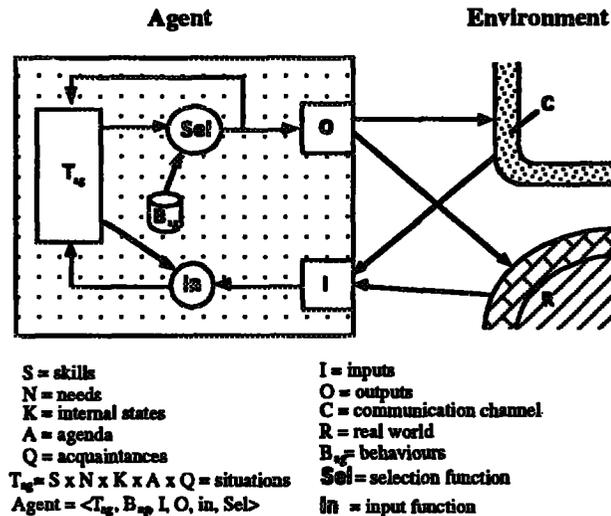


Figure 1: The Cooperative Architecture Model.

Although these principles derive from a particular experience, they can be generalised and used as general guidelines for designing and implementing a modular and flexible architecture for cooperative interaction of agents. Moreover, the underneath agent model results to be sufficiently general to be adopted in various domains and applications.

References

Sommaruga, L., Avouris, N., Van Liedekerke, M. 1995. The Evolution of the CooperA Platform. Chapt. 14. In O'Hare, G., and Jennings, N., eds. 1995. *Foundations of Distributed Artificial Intelligence*, Wiley Inter-Science, Sixth-Generation Computer Technology Series.